

船舶小组立装焊多智能体鲁棒调度*

闫江^{1a},张亚辉^{1b,2},胡小锋^{1a,2}

(1.上海交通大学 a.机械与动力工程学院;b.海洋装备研究院,上海 200240;2.上海长兴海洋实验室,上海 201913)

摘要:针对船舶小组立装焊调度计划抵抗干扰能力不足的问题,以最大化鲁棒性和最小化最大完工时间为目标,提出一种引入数据交互机制的多智能体鲁棒调度算法。首先,构建了多约束下的数学模型;其次,针对工序选择和缓冲区配置两个子问题,设计了缓冲区智能体和规则选择智能体的结构、状态空间、动作空间和奖励函数,在近端策略优化算法基础上,引入智能体间的数据交互机制,提升算法的收敛效果;最后经过案例验证,所提算法的求解性能普遍优于对比算法,证明了所提算法的有效性。

关键词:船舶小组立装焊;鲁棒调度;强化学习;多智能体

中图分类号:TH165;TG659 **文献标识码:**A

Multi-Agent Robust Scheduling of Ship Group Assembly Welding

YAN Jiang^{1a}, ZHANG Yahui^{1b,2}, HU Xiaofeng^{1a,2}

(1a. School of Mechanical Engineering; b. Institute of Marine Equipment, Shanghai Jiaotong University, Shanghai 200240, China; 2. Shanghai Changxing Ocean Laboratory, Shanghai 201913, China)

Abstract: Aiming at the problem of insufficient resistance to interference of ship group assembly welding scheduling plan, with maximizing robustness and minimizing maximum completion time as the optimization goal, a multi-agent robust scheduling algorithm introducing data interaction mechanism is proposed. First, A mathematical model under multiple constraints is constructed. Secondly, the structure, state space, action space and reward functions of buffer area agent and rule selection agent are designed for the sub-problems of process selection and buffer allocation. Based on the proximal policy optimization, the data interaction mechanism between agents is introduced to improve the convergence effect of the algorithm. Finally, through case verification, the performance of the proposed algorithm is generally better than that of the comparison algorithm, which proves the effectiveness of the proposed algorithm.

Key words: ship group assembly welding; robust scheduling; reinforcement learning; multi-agent

0 引言

小组立是船舶制造的最基本结构单元,具有数量众多、工期紧张的特点。小组立生产需按照装焊计划执行,但作业现场易发生因工人技能熟练度不足导致工序作业延时的扰动,使得作业难以按照初始计划执行,往往造成拖期的后果,降低了船舶制造效率。在项目计划的制定中,鲁棒性是指系统在不确定因素下仍然能够保持正常工作的特性^[1]。小组立装焊计划抵抗干扰能力差的现状,反映出装焊计划鲁棒性不足的问题,因此本文开展小组立船舶装焊调度鲁棒性计划的研究。

目前,小组立装焊调度算法以重调度或在线调度

为主^[2-5],没有考虑计划的鲁棒性。但发生扰动后,未考虑鲁棒性的生产计划在多次修改和大幅度变动下易导致实际执行中出现混乱。崔南方等^[6]指出,项目执行过程中存在的不确定因素增加了基准调度计划如期执行的难度,采用时间缓冲区技术构建鲁棒调度计划是解决该问题的主要方法。针对鲁棒调度问题,马萍等^[7]提出结合灰色关联分析法的缓冲区鲁棒算法。周尧尧等^[8]提出基于综合资源约束的关键链鲁棒算法,设置有效缓冲区来保护项目按计划完工。VAN等^[9-10]以分散缓冲思想为基础提出了RFDF(resource flow dependent float factors)以及STC(starting time criticality)鲁棒调度方法。胡振涛等^[11]设计了集成时间缓冲

收稿日期:2024-10-25;修回日期:2024-12-01

*基金项目:国家自然科学基金青年科学基金项目(52405558);上海市科学技术委员会项目(23XD1450400)

作者简介:闫江(1999—),男,硕士研究生,研究方向为船舶组立装焊调度算法,(E-mail)yanjiang121@sjtu.edu.cn;通信作者:胡小锋(1977—),男,研究员,博士生导师,研究方向为运筹学的工程应用,(E-mail)wshxf@sjtu.edu.cn。

引用本文:闫江,张亚辉,胡小锋.船舶小组立装焊多智能体鲁棒调度[J].组合机床与自动化加工技术,2025(9):195-200.

YAN Jiang, ZHANG Yahui, HU Xiaofeng. Multi-agent robust scheduling of ship group assembly welding[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2025(9):195-200.

与资源流的多技能项目鲁棒调度算法。以上研究只关注了缓冲区的配置,提高了项目的鲁棒性但以牺牲时间效率来实现。然而实际中项目管理者不仅注重计划的鲁棒性,也关注最大完工时间等指标,因此本文以最小化最大完工时间与最大化鲁棒性作为优化目标,考虑计划鲁棒性和计划时间效率两方面的权衡。

多智能体方法具有自学习生成调度策略的特点,适用于解决复杂的车间项目调度问题。李小涛等^[12]提出了遗传算法与模拟退火算法的混合多智能体算法;刘亚辉等^[13]提出了双系统驱动的双环深度 Q 网络方法来解决柔性车间调度问题;李健等^[14]针对工件与机器的同时选择问题,提出协同智能体强化学习方法。然而小组立装焊调度作为典型的复杂项目调度问题,基于多智能体的算法尚未被用于解决该问题。

综上所述,如何设计求解小组立装焊鲁棒调度问题的多智能体,并使用该多智能体算法获得兼具更高鲁棒性和更短完工时间的小组立装焊调度计划,是本文的研究关键。本文构建了多约束下的鲁棒调度数学模型,提出一种引入数据交互机制的多智能体鲁棒调度算法(multi-agent robust scheduling algorithm introducing data interaction mechanism, MARSA-DIM)进行模型求解,最后通过实验证明了所提算法的有效性。

1 船舶小组立装焊鲁棒调度模型

1.1 问题描述

小组立装焊过程如图 1 所示,首先将多块筋板焊接到底板上成为肋板,然后将多块肋板与纵桁焊接形成小组部件。装焊计划的制定需考虑多种约束。装焊中工序作业延时的扰动时有发生,为削减其对计划执行的影响,基于分散缓冲的思想为各工序分配缓冲时间,使工人拥有额外时间继续完成工序。

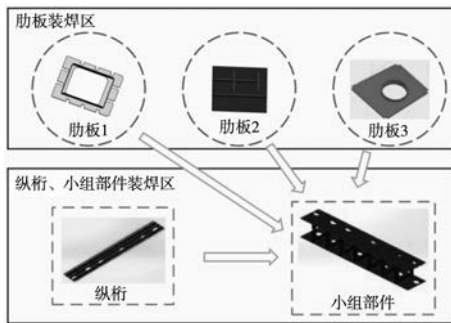


图 1 小组立装焊过程

本文中一个肋板、一个纵桁以及一个小组部件的装焊分别称为一个任务组,每个任务组包括 5 道工序。小组立鲁棒装焊调度问题可总结为:在满足工序先序约束、工人资源约束和场地资源约束的条件下,为小组立的工序配置缓冲时间以抵抗工序作业延迟的干扰,生成一个鲁棒的小组立装焊调度计划,目标为最小化最大完工时间和最大化鲁棒性。问题存在以下假设:

(1) 装焊场地为有限矩形区域,无高度限制,零件的摆放不能重叠,装焊完成后占有区域会被释放。

(2) 工序需要的工种和工人数量不同。

(3) 零件的摆放位置一旦确定,在该零件作业完成移除场地之前,摆放位置不会改变。

(4) 零件在装焊场地上占用的区域为零件投影平面的最小外接矩形。

1.2 数学模型

本文涉及的符号定义如下: i 为任务组编号; t 为模拟时钟, $t=0,1,2,\dots$; G_{ij} 为第 i 个任务组的第 j 个工序; G_{ijw} 为工序 ij 所需的焊接工人数量; G_{ijp} 为工序 ij 所需的打磨工人数量; W_w 为焊接工人总数; W_p 为打磨工人总数; TS_{ij} 为工序 ij 的开始时间; TL_{ij} 为工序 ij 的工时; TB_{ij} 为工序 ij 的缓冲区时间; s_t^i 为 t 时刻任务组 i 所占的平面场地; s 为场地区域; S 为不同场地的集合; T_{\max} 为所有工序的最大结束时间; $T_{\max}(G_{ij})$ 为工序 G_{ij} 的所有紧前工序中的最大结束时间; N_t 为 t 时刻的工序集合; w_b^t 为 t 时刻的 b 工序所需的焊接工人数量; p_b^t 为 t 时刻的 b 工序所需的打磨工人数量; t_{\max} 为已调度工序中的最大时间; G_{done} 为已调度的工序数量; G_{undo} 为未调度的工序数量; $path$ 为关键路径长度。

为抵抗作业过程中的干扰需提升计划鲁棒性,因此设置最大化鲁棒性为优化目标,如式(1)所示,本文在优化阶段对调度计划的鲁棒性使用调度计划的缓冲区时间总和来衡量。最小化最大完工时间为项目管理的关键目标,且过多增加缓冲时间 TB_{ij} 会导致计划最大完工时间的延长,因此设置最小化最大完工时间为优化目标,如式(2)所示。

$$\max(\sum_i \sum_j TB_{ij}) \quad (1)$$

$$\min(T_{\max}) \quad (2)$$

(1) 工序先序约束:后一道工序必须在所有紧前工序完成且经过缓冲时间后才能开始。即工序的开始时间应晚于该工序的所有紧前工序的最大结束时间与该工序的缓冲时间之和。

$$TS_{ij} \geq T_{\max}(G_{ij}) + TB_{ij} \quad (3)$$

(2) 工人资源约束:小组立作业工人的数量有限。任意时刻正在作业的所有工序所需的某工种工人数量要小于该工种工人总数。

$$\sum_b w_b^t \leq W_w, b \in N_t, t = 0, 1, 2, \dots \quad (4)$$

$$\sum_b p_b^t \leq W_p, b \in N_t, t = 0, 1, 2, \dots \quad (5)$$

(3) 场地资源约束:小组立钢板的摆放不能超出装焊场地或重叠。任意时刻正在作业的工序所占的场地区域 s_t^i 是整体区域 s 的子集,任意时刻正在作业的不同工序所占的场地区域不重合。

$$s_t^i \subseteq s, t = 0, 1, 2, \dots \quad (6)$$

$$s_t^i \cap s_{t'}^{i'} = \emptyset, i \neq i', t = 0, 1, 2, \dots \quad (7)$$

2 小组立装焊鲁棒调度问题求解

小组立装焊鲁棒调度问题可分为任务先序选择、缓冲区时间配置两个子问题。针对以上子问题分别设计工序选择智能体(rules selection agent, RSA)、缓冲区智能体(buffer area agent, BAA),智能体与环境的交互采用马尔可夫决策过程进行描述。

2.1 多智能体结构和数据交互机制

近端策略优化(proximal policy optimization, PPO)算法是本文单个智能体的基线算法,在车间调度问题中被广泛运用。多智能体结构如图2所示,左半侧为RSA智能体,右半侧为BAA智能体,每个智能体内部为近端策略优化算法的动作网络和评价网络。

目前,使用多智能体强化学习解决车间调度问题中,传统方法是仅采用车间环境的基础特征 $state_0$ 例如工人利用程度、已调度工序百分比来构建智能体观测的状态空间^[15-16]。但车间调度中多智能体之间进行决策的一个显著特点是异步进行且关联性强,例如 X 智能体对 x 问题进行决策后, Y 智能体才可对 y 问题进行决策,直至完成调度。传统方法并没有将智能体间的动作信息以及动作产生的附加结果加入状态,从而忽视了智能体间动作先后之间的关联影响产生的状态数据,这会使智能体在训练中面临数据信息不足带来的收敛速度慢、获取目标奖励值低的问题,在本文问题中不利于生成鲁棒性高且兼顾时间性能的计划。例如BAA智能体配置缓冲时间不仅需要基本环境信息,也要考虑到RSA智能体选择的工序的时长、所需资源;RSA智能体选择工序也需要考虑上一步BAA智能体配置的缓冲区时长。

为充分利用智能体间交互信息,在智能体间引入数据交互机制,如图2中虚线所示为数据交互路径。RSA智能体在第 t 步做出动作 a_{rsa} 后,根据 a_{rsa} 匹配调度规则选出工序 b 。随后将 a_{rsa} 和工序 b 的附加信息拼接为向量 $state_2 = [a_{rsa}, s_4, s_5, s_6]$,与 $state_0$ 共同传递为BAA智能体的状态参数。第 $t+1$ 步BAA智能体做出动作 a_{baa} 后,根据 a_{baa} 配置工序 b 的缓冲时间 TB 。随后将 a_{baa} 和选择出的缓冲区信息拼接为向量 $state_1 = [a_{baa}, s_2]$,与 $state_0$ 共同作为RSA智能体的状态参数。在调度的起点,RSA智能体需要从工序集合中选出第一个工序进行调度,此时BAA智能体还没有向RSA智能体传递 $state_1$ 向量,因此交互起点的 $state_1$ 初始化为零向量,后续两个智能体则可有序进行交互。

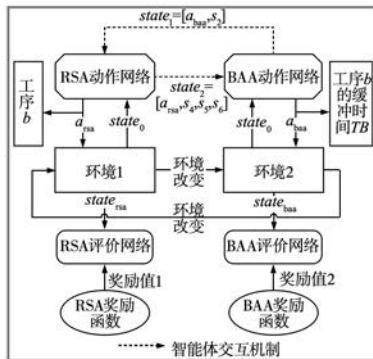


图2 Robust-multi-agent 模型结构和交互机制

2.2 状态空间设计

状态空间向量是智能体动作网络和评价网络的输入,RSA智能体的状态空间 $state_{rsa} = [state_0, state_1]$,BAA智能体的状态空间 $state_{baa} = [state_0, state_1]$,如表

1所示。 $state_0$ 是车间环境基础特征信息, $state_1$ 和 $state_2$ 分别是RSA智能体和BAA智能体在数据交互机制下的状态信息。

表1 智能体状态空间参数表

参数类型	表达式	含义
车间基础 状态参数 $state_0$	$s_w = \frac{\sum C_{jw}}{t_{max}}$	焊接工人利用程度
	$s_p = \frac{\sum G_{jw}}{t_{max}}$	打磨工人利用程度
	$s_d = \frac{C_{done}}{t_{max}}$	已调度工序数量/ 已调度工序中的最大时间
	$s_c = \frac{t_{max}}{path}$	已调度工序中的最大时间/ 关键路径长度
	$s_{nd} = \frac{C_{undo}}{path}$	未调度工序数量/关键路径长度
	RSA 智能体 交互参数 $state_1$	$s_1 = a_{baa}$
$s_2 = TB_j$		缓冲区时长
$s_3 = a_{rsa}$		RSA 智能体动作
BAA 智能体 交互参数 $state_2$	$s_4 = TL_j$	工序作业时长
	$s_5 = G_{jw}$	工序的焊接工人数量
	$s_6 = G_{jp}$	工序的打磨工人数量

2.3 动作空间

动作空间是智能体的动作集合,RSA智能体的动作空间为启发式调度规则集合,设计如下:①选择最短加工时间的工序(SPT);②选择最长加工时间的工序(LPT);③选择最高优先级的工序(HPF);④选择工人数目之和最大的工序(MAXWN);⑤选择工人数目之和最小的工序(MINWN)。

本文工时范围在2~10,工时为2、3、4的工序约占总数的90%,通常缓冲区时间小于等于工时并与工时相关,因此设计BAA智能体动作空间如下:①配置0单位缓冲;②配置1单位缓冲;③配置2单位缓冲;④缓冲为工时的一半并取整;⑤缓冲区等于工时。其中规则①、②、③为固定值动作,规则④、⑤与工时大小相关。

2.4 奖励函数

强化学习的奖励函数应与优化目标具有一致性,设计RSA智能体进行最小化最大完工时间的优化,BAA智能体进行最大化鲁棒性的优化。如果仅将奖励值定为最大完工时间的负数,会造成稀疏奖励的问题。调度过程中资源利用程度的增加会使最大完工时间降低,因此RSA智能体的奖励函数设置为:

$$r_{rsa} = \begin{cases} s_{wp}, s_{wp} \geq s'_{wp}, & done = 0 \\ 0, s_{wp} < s'_{wp}, & done = 0 \\ -1.5 \times T_{max}, & done = 1 \end{cases} \quad (8)$$

$done$ 为0表示全部工序还未调度完成, s_{wp} 指的是当前调度步骤下已经调度完成的工序中,两种工人资源的利用程度之和。如果当前步骤的 s_{wp} 大于等于上一调度步骤,说明当前步骤的调度使已生成的部分调度计划资源利用程度不变或上升,给予 s_{wp} 大小的奖励;否则给予0奖励。 $done$ 为1表示全部工序已调度完成,给予最大完工时间的-1.5倍奖励。

BAA智能体的奖励函数设置如下:

$$r_{baa} = \begin{cases} 0, & TB_{ij} = 0 \\ 0.2, & TB_{ij} = 1 \\ 0.4, & TB_{ij} = 2 \\ 0.2 \times \frac{TL_{ij}}{2}, & TB_{ij} = \frac{TL_{ij}}{2} \\ 0.2 \times TL_{ij}, & TB_{ij} = TL_{ij} \end{cases} \quad (9)$$

式中： TB_{ij} 是指每个工序配置的缓冲区时长，奖励函数值为 TB_{ij} 的 0.2 倍，激励 BAA 智能体增加缓冲时间的配置。

当 BAA 智能体为计划配置缓冲时间 TB_{ij} 逐渐增多导致 T_{max} 增大后，RSA 智能体的 $-1.5 \times T_{max}$ 则形成的负奖励会增多，抑制 BAA 智能体缓冲时间配置的增多，从而有利于 T_{max} 的减小，系数 -1.5 可在训练前调节，若减小系数则代表管理者在两个目标中更为重视最小化最大完工时间的目标。

2.5 算法步骤

MARSA-DIM 算法的多智能体与装焊车间环境不断交互生成鲁棒调度计划，在调度过程中记录调度信息。当组立订单的全部工序调度完成后，得到调度轨迹。

采用多目标优化中帕累托非劣解的思想，训练中当前回合新生成的计划若在两个优化目标 X_1 、 X_2 上的表现不同时差于上一回合计划的两个优化目标 X'_1 、 X'_2 ，则正常更新；若新计划在两个优化目标上的表现都更差，则智能体的学习率被调整至初始学习率的 0.2 倍执行更新，减少对较差解的学习。通过奖励函数的设计和上述参数更新方法进行时间效率和鲁棒性两个目标的权衡。

将其作为训练数据对网络进行更新，步骤为：

步骤 1：随机初始化 RSA 智能体网络参数 θ_{rsa} 、 φ_{rsa} 和 BAA 智能体网络参数 θ_{baa} 、 φ_{baa} ，设置训练参数、训练次数和训练批次；

步骤 2：RSA 智能体获取调度状态 $state_{rsa}$ ，依据策略 $\pi_{rsa}(a_{rsa} | state_{rsa})$ 选择动作 a_{rsa} ，执行得到奖励 r_{rsa} ；

步骤 3：BAA 智能体获取调度状态 $state_{baa}$ ，依据策略 $\pi_{baa}(a_{baa} | state_{baa})$ 选择动作 a_{baa} ，执行得到奖励 r_{baa} ；

步骤 4：智能体网络计算网络损失并更新参数；

步骤 5：训练结束后，在考虑各种约束的基础上，使用两个智能体的动作网络对验证案例调度生成鲁棒计划。

智能体训练参数如表 2 所示。

表 2 多智能体算法训练参数

参数	值
actor _{rsa} 神经元个数	(7,64,16,5)
actor _{baa} 神经元个数	(9,64,16,5)
critic _{rsa} 神经元个数	(7,64,16,1)
critic _{baa} 神经元个数	(9,64,16,1)
actor 输出层激活函数	Softmax
隐藏层激活函数	Relu
初始学习率	0.001
折扣因子	0.9

3 实验结果与分析

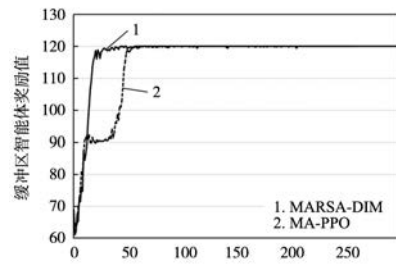
3.1 案例说明

本文的案例来自于文献[4]中的组立装焊数据，

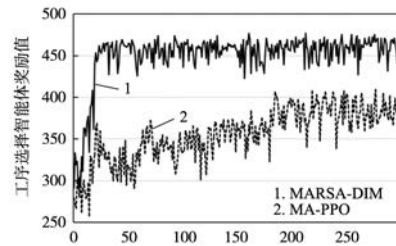
肋板装焊区面积为 28 m × 12 m，小组部件装焊区面积为 14 m × 12 m，班组拥有装焊工 19 人，打磨工 20 人。部件所占的场地区域以其外接矩形代替，小组立空间配置算法采用矩形顶点搜索法。150 道工序的案例共有 30 个任务组，90 道工序的案例共有 18 个任务组。

3.2 训练结果

本文采用 Python3.7 编程，操作系统为 Windows 10，CPU 为 Intel(R) Core(TM) i7-7700，频率为 3.60 GHz，内存 16 GB。算法使用 60 个小组立数据作为训练集编程实现。经参数设置后，将 MARSA-DIM 算法进行训练，同时与未引入数据交互机制的仅使用 $state_0$ 作为状态的多智能体近端策略优化算法 (MA-PPO) 对比，最后两种算法的奖励值收敛如图 3 所示。



(a) BAA 智能体奖励值



(b) RSA 智能体奖励值

图 3 模型训练过程奖励收敛曲线

图 3a 中，两个算法的 BAA 智能体的最终奖励一致，但 MA-PPO 算法在约 25 ~ 50 回合奖励上升趋势平缓，说明算法难以寻找到更优的策略增加计划的缓冲时间，相比之下 MARSA-DIM 算法的迭代过程更迅速，在约 30 回合达到了稳定。图 3b 中，MA-PPO 算法在约 25 ~ 50 回合在最大完工时间的优化上奖励值出现了下降趋势，而后在 200 回合的训练后奖励值才逐步稳定至 370 附近，相比之下 MARSA-DIM 算法的 RSA 智能体迭代过程上升更平稳快速，且奖励值高于对比算法。综上引入数据交互机制的改进算法使智能体在训练中获取其余智能体的动作和动作附加信息，加快了奖励函数的收敛速度，使智能体学习到了更优的策略。

3.3 实验对比

使用 MARSA-DIM 算法生成的小组立装焊计划如图 4 所示。可以看出算法为每个工序都配置了缓冲，其中工时较长的工序缓冲时间配置更多，例如工时为 9 的 25-2 工序配置了 5 个时间单位的缓冲；处于关键路径上的 26、30 任务组的工序缓冲配置也更长。工序

4-1、8-1 等开工时间较晚是空间场地约束以及工人资源约束导致;工序 27-1、28-1 等开工较晚是工序先序约束导致。

为验证 MARSА-DIM 算法对最大完工时间和鲁棒性的优化效果,在 90 道和 150 道工序的两组案例上,与常用启发式规则 (SPT、HPF)、近端策略优化算法 (PPO)、遗传算法 (genetic algorithm, GA) 与缓冲区配置算法结合进行比较,近端策略优化算法使用的是经过时间优化训练后的神经网络模块,遗传算法是以最小化最大完工时间为目标。其中启发规则生成计划的缓冲区是随机缓冲配置算法 (random algorithm) 在 BAA 智能体动作空间随机选择生成。近端策略优化算法和遗传算法生成计划的工序缓冲区按缓冲区配置的经典方法剪切法 (cut and paste method, C&PM)^[17] 的思想进行设定。

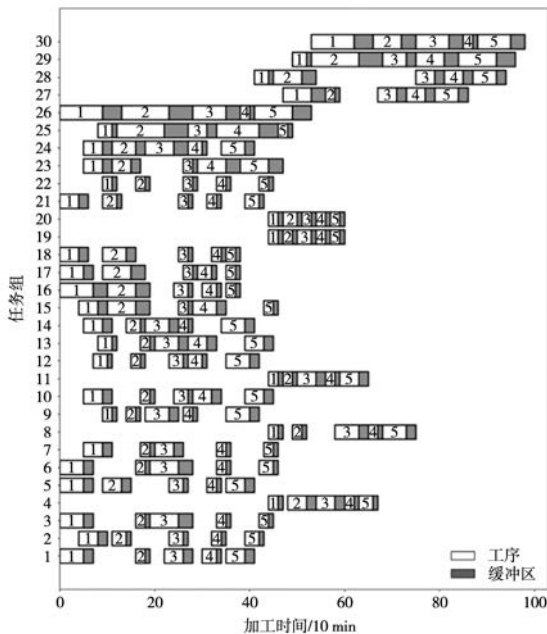


图 4 小组立装焊调度甘特图

最大完工时间的比较结果如表 3、表 4 所示,可知 MARSА-DIM 算法生成的调度计划在最大完工时间上普遍低于对比算法,在时间性能上高于启发式规则下的随机缓冲算法 5% 以上,相比于与剪切法结合的近端策略优化算法、遗传算法也高出 1.5% 以上。

表 3 最大完工时间

算法	90 道工序案例					
	1	2	3	4	5	6
MARSА-DIM	70	72	68	68	64	75
SPT + Random	90	61	63	66	71	89
HPF + Random	87	67	89	89	65	85
GA + C&PM	73	72	75	70	64	72
PPO + C&PM	75	78	73	70	67	75

算法	150 道工序案例					
	1	2	3	4	5	6
MARSА-DIM	101	99	97	107	103	106
SPT + Random	112	102	113	103	119	116
HPF + Random	113	106	114	106	108	117
GA + C&PM	103	101	105	109	102	103
PPO + C&PM	106	108	109	110	106	109

表 4 MARSА-DIM 算法的时间性能提升百分比 (%)

算法	90 道工序案例 提升百分比	150 道工序案例 提升百分比
SPT + Random	5.2	7.8
HPF + Random	13.4	7.6
GA + C&PM	2.1	1.5
PPO + C&PM	4.8	5.4

计划的鲁棒性主要采用各种鲁棒性指标来衡量,本文选取 5 种鲁棒性指标如下:

$$R_1 = \frac{\sum_i \sum_j TB_{ij}}{T_{\max}} \quad (10)$$

$$R_2 = \sum_i \sum_j TB_{ij} \quad (11)$$

$$R_3 = \frac{\sum_i \sum_j n}{\sum_i \sum_j 1}, n = \begin{cases} 0, & TB_{ij} = 0 \\ 1, & TB_{ij} \neq 0 \end{cases} \quad (12)$$

$$R_4 = \frac{\sum_{i=1}^N \sum_{ij} |TS_{ij} - ts_{ij}|}{N} \quad (13)$$

$$R_5 = \frac{1}{T_{\max}} \sum_{i=1}^{T_{\max}} (x_i - \mu)^2 \quad (14)$$

式中: R_1 是全部工序缓冲区时间之和与完工时间的比值,衡量了单位时间内计划拥有的缓冲时间大小; R_2 是全部工序缓冲区时间之和, R_3 是缓冲区时间不为 0 的工序数目与全部工序数目的比值, R_4 衡量了计划的抗扰动恢复能力,设定每个工序作业时以 0.1 的概率发生作业延时扰动,扰动时长为工时的 0.5 倍,对每个工序的计划开始时间和实际开始时间差取绝对值求和后取均值; N 是执行模拟的次数,重调度方案为右移重调度;由于工人资源是装焊所需最多且常为瓶颈的资源,使用工人资源作为资源均衡性的评价指标, R_5 为计划全局使用工人总数的方差, x_i 为 i 时刻下使用的工人总数量, μ 为使用工人总数的均值。

对于 R_1 、 R_2 、 R_3 指标,将 6 个案例的鲁棒性指标求解并取均值;对于 R_4 指标,取第一个案例,模拟 100 次后取均值,结果如表 5、表 6 所示。

表 5 算法鲁棒性指标性能对比

算法	90 道工序案例			
	R_1	R_2	R_3	R_4
MARSА-DIM	1.90	132.0	1.00	41.2
SPT + Random	1.52	111.2	0.80	60.3
HPF + Random	1.41	112.6	0.77	72.7
GA + C&PM	1.92	136.5	1.00	38.9
PPO + C&PM	1.87	136.5	1.00	46.6

算法	150 道工序案例			
	R_1	R_2	R_3	R_4
MARSА-DIM	2.43	248.2	0.97	72.6
SPT + Random	1.75	194.2	0.79	103.0
HPF + Random	1.83	202.6	0.79	117.5
GA + C&PM	2.30	239.2	1.00	85.8
PPO + C&PM	2.21	239.2	1.00	80.2

相比于“启发式规则 + 随机缓冲算法”,MARSА-DIM 算法在前 4 个鲁棒性指标上均优于对比算法 20% 左右, R_1 高于对比算法说明本文算法生成的计划单位时间内拥有的缓冲时间更多。 R_2 更高说明算法能够为计划配置更多的缓冲时间。 R_3 更高且配置率

趋近于 1.0,表明本文算法倾向于为所有工序配置缓冲时间,缓冲时间的配置更均匀。 R_4 指标优于对比算法说明计划在扰动影响下经过更少的时间调整即可恢复运行。

表 6 MARSА-DIM 算法的鲁棒性提升百分比 (%)

算法	90 道工序案例			
	R_1	R_2	R_3	R_4
SPT + Random	25.7	18.7	25.0	31.6
HPF + Random	34.8	17.2	29.8	43.3
GA + C&PM	-1.0	-3.3	0.0	-5.9
PPO + C&PM	1.6	-3.3	0.0	11.6
算法	150 道工序案例			
	R_1	R_2	R_3	R_4
SPT + Random	38.9	27.8	26.5	29.5
HPF + Random	32.8	22.5	26.5	38.2
GA + C&PM	5.7	3.8	-3.0	18.2
PPO + C&PM	10.0	3.8	0.0	9.5

相比于“遗传算法 + 剪切法”和“近端策略优化 + 剪切法”,在 90 道工序案例上本文算法比对照算法的 R_1 、 R_2 、 R_4 指标普遍略低但相近,在 150 道更大规模案例上本文算法的 R_1 、 R_2 和 R_4 指标表现更好。MARSА-DIM 在 R_3 指标上与融合了剪切法的对比算法相近,原因是使用的剪切法会为所有的工序配置缓冲区,本文算法已经达到近 1.0 的较高配置率。

对于 R_5 指标,越小说明资源的分配越均衡,针对第一组案例求解,结果如图 5 所示。发现 MARSА-DIM 算法对于其他算法优势不突出,与图 4 结合分析,是因为算法训练中用提高资源利用率作为手段优化了最大完工时间,使得生成的计划前期 0~40 时段工序较多资源占用增大,中期 40~80 时段的资源分配变少。考虑到缓冲区中的资源冗余对于缓解资源配置不够均衡带来的风险是有益的,因此尽管 MARSА-DIM 算法在资源分配均衡性的表现并不突出,但其在时间优化和其余鲁棒性指标方面的优势仍然值得肯定。

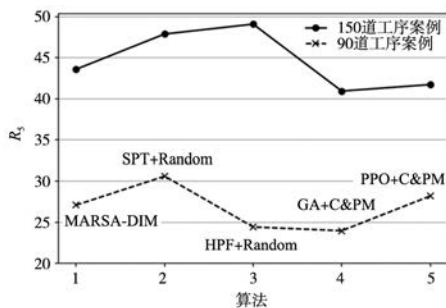


图 5 不同算法 R_5 指标比较

4 结论

本文进行了小组立装焊鲁棒调度计划的研究,以最大化调度计划鲁棒性和最小化最大完工时间为优化目标,构建了多约束下的小组立鲁棒调度数学模型,提出了引入数据交互机制的多智能体鲁棒调度算法进行求解。算法改进前后的奖励收敛曲线表明 MARSА-DIM 算法的迭代收敛相比 MA-PPO 算法更稳定,奖励值更高。算法测试结果显示本文算法生成的计划在最大完工时间和多个鲁棒性指标上普遍优于对比算法,

说明算法生成的调度计划在兼顾了时间性能的基础上增加了鲁棒性,利于抵抗扰动。本文算法为组立装焊鲁棒计划的制定提供了一种可行方案,考虑资源均衡配置将是后续研究的重点内容。

[参考文献]

- [1] NEWMAN M E. The structure and function of complex networks[J]. SIAM Review, 2003, 45(2): 167-256.
- [2] 李磊,王晨,周宏根,等. 面向多约束的船舶组立构件建造作业调度方法[J]. 船舶工程, 2020, 42(3): 135-140.
- [3] 侯金伟,胡小锋,徐昇. 多规则融合的船体小组立部件装焊作业调度算法[J]. 船舶工程, 2020, 42(5): 101-107.
- [4] 张澳圆,胡小锋,张亚辉. 多场景多目标动态变化下船舶小组立装焊重调度[J]. 上海交通大学学报, 2025, 59(4): 476-488.
- [5] 陈冶,景旭文,刘磊,等. 知识驱动的船舶分段建造车间在线调度方法[J]. 船舶工程, 2023, 45(7): 156-161.
- [6] 崔南方,赵雁,胡雪君. 鲁棒性项目调度中缓冲设置方法[J]. 控制与决策, 2014, 29(2): 368-372.
- [7] 马萍,樊燕燕,魏兴华,等. 基于改进灰色关键链的缓冲设置方法研究[J]. 工程管理学报, 2020, 34(1): 149-153.
- [8] 周尧尧,刘猛锐,朱训国,等. 基于综合资源约束的关键链缓冲区大小计算方法[J]. 土木工程与管理学报, 2020, 37(6): 145-151.
- [9] VAN D V, DEMEULEMEESTER E, HERROELEN W, et al. The trade-off between stability and makespan in resource-constrained project scheduling [J]. International Journal of Production Economics, 2006, 44(2): 215-236.
- [10] VAN D V, DEMEULEMEESTER E, HERROELEN W. Proactive heuristic procedures for robust project scheduling: an experimental analysis [J]. European Journal of Operational Research, 2008, 189(3): 723-733.
- [11] 胡振涛,崔南方,胡雪君,等. 集成时间缓冲与资源流的多技能项目鲁棒调度方法[J]. 系统管理学报, 2023, 32(5): 916-926.
- [12] 李小涛,彭翀. 基于混合多智能体遗传算法的作业车间调度问题研究[J]. 北京航空航天大学学报, 2017, 43(2): 410-416.
- [13] 刘亚辉,申兴旺,顾星海,等. 面向柔性作业车间动态调度的双系统强化学习方法[J]. 上海交通大学学报, 2022, 56(9): 1262-1275.
- [14] 李健,李涸坤,何鹏博,等. 协同智能体强化学习算法的柔性作业车间调度方法研究[J]. 系统仿真学报, 2024, 36(11): 2699-2711.
- [15] 孟繁威,郭宏,延小龙,等. 基于多智能体强化学习求解柔性作业车间联合调度问题[J/OL]. 计算机集成制造系统. <https://doi.org/10.13196/j.cims.2024.0236>.
- [16] 王无双,骆淑云. 基于强化学习的智能车间调度策略研究综述[J]. 计算机应用研究, 2022, 39(6): 1608-1614.
- [17] STEYN H. An investigation into the fundamentals of critical chain project scheduling [J]. International Journal of Project Management, 2001, 19(6): 363-369.

(编辑 祝智铭)